UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,543	12/30/2003	**Frank Kilian**	6570P012	8850

45062 7590 07/09/2008
SAP/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
CAO, DIEM K	
ART UNIT	PAPER NUMBER
2194	
MAIL DATE	DELIVERY MODE
07/09/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/749,543	**Applicant(s)** KILIAN, FRANK	
	Examiner DIEM K. CAO	**Art Unit** 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 April 2008*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.
4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application
6) □ Other: _____.

DETAILED ACTION

1. Claims 1-39 are pending.

2. In view of the Appeal Brief filed on 4/30/2008, PROSECUTION IS HEREBY REOPENED. A new ground of rejection is set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-12 and 34-35 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the

relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 1 recites "a communication interface coupled between the launch logic and the control logic to enable the launch logic to obtain status of each of the Java processes and enable the control logic to access the status in a shared memory via the communication interface, the launch logic to store and maintain the status in the shared memory via the communication interface", however, the specification does not support the above limitation. The specification seems to disclose to enable the control logic to communicate with the launch logic, a communication interface mechanism is set up, the communication interface mechanism may be based on any suitable communication mechanism such as shared memory, pipes, queues, signals (page 4, paragraph [00017]). The launch logic using Java Native Interface to obtain information regarding status of each of the Java processes and update the shared memory with the information relating the status of the Java processes (page 5, paragraph [00018]). The control logic may establish a communication with the shared memory via a suitable communication interface 315, such as a C-library to obtain the information regarding the status of the processes (page 5, paragraphs [00019] – [00020]) . Clearly, the launch logic and the control use different communication interfaces to access the shared memory, not a single one as claimed in the above limitation.

Claims 2-12 and 34-35 fail to remedy the deficiencies of 1 above, and therefore are rejected under the same ground of rejection.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Matena et al (US PGPub: 2005/0005200), hereinafter Matena, in view of Sohda et al. (Implementation of a Portable Software DSM in Java).

As to Claim 1, Matena teaches the invention substantially as claimed including a system comprising:

a) a cluster having a first instance and a second instance, each of the first and second instances including a plurality of server nodes (para. [0078], [0083]);

b) a control logic (execution controller) to start each instance by initiating a launch logic for each of the server nodes, the launch logic (application controller or Java Application controller; page 14, paragraph 188 and page 15, paragraphs 208-209), when initiated, to execute Java processes in each respective server node (para. [0114], [0123]-[0132], and [0413]); and

c) a communication interface coupled between the launch logic and the control logic (Java Application Controller Application Programming Interface; page 16; paragraph 221) to enable the launch logic to obtain status of each of the Java processes (The "stop application" … the "obtain application information" 2410 operations return status information about the running application; page 16, paragraphs [0210]-][0214]) and enable the control logic to access the status in memory via the communication interface (The JAC API 2430 allows the system management tool and other components to manage the lifecycle of application … to its users; page 16, paragraph [0221]), the launch logic to store and maintain the status in the memory via the communication interface (JAC 2401 maintains … belong to this container group; page 16, paragraphs [0216]-[0218]).

Matena does not explicitly teach shared memory. The only different between this claim and the system of Matena is that in Matena, the control logic obtains the status of the processes using message passing, wherein in this claim, the control logic obtains the status of the processes using shared memory. However, Sohda teaches shared memory between nodes in clusters that implemented in Java (Abtract and pages 165-167, sections 2.2 "Summary of JSDM Implementation Choices/Policies", section 3 "JDSM Implementation and section 3.2 "DSM Runtime Implementation").

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Matena with the teachings of Sohda by using shared memory technique instead of message passing because of all well-known advantages of using shared memory over message passing in the art, and both the control logic and launch logic can access and update the status of the processes concurrently. Furthermore, using shared memory or message passing are design choice.

As to Claim 2, Matena further teaches wherein the launch logic is provided to load a virtual machine and execute a Java process in the virtual machine (para. [0193]-[0196]).

As to Claim 3, Matena as modified by Sohda further teaches wherein the communication interface comprises: a shared memory to store the status of the Java processes (see Matena: JAC 2401 maintains … belong to this container group; page 16, paragraphs [0216]-[0218]) and (discussion in claim 1 above regarding Sohda teaching of shared memory).

As to Claim 4, Matena does not explicitly teach wherein the launch logic comprises a Java native interface to obtain the status of each of the Java processes and to update the shared memory with the obtained status.

However, Sohda teaches the launch logic comprises a Java native interface to obtain the status of each of the Java processes and to update the shared memory with the obtained status (The Server itself is composed ... interfaces via JNI; pages 165-166, section 3 "JDSM Implementation").

It would have been obvious to one of ordinary skill in the art at the time of invention to apply the teaching of Sohda to the system of Matena because to access memory of the system to obtain the status of the processes/applications, Java Virtual Machine needs to utilize the Java Native Interface to invokes modules/functions to obtain those information that supports by the systems/OS that the JVM runs on.

As to Claim 5, Matena as modified by Sohda teaches the control logic accesses the shared memory to monitor the status of each of the Java processes (page 16, paragraph [0221]).

As to Claim 6, <u>Matena</u> further teaches wherein the control logic is provided to detect a failure of a Java process and to automatically restart the failed Java process (para. [0242]-[0248]).

As to Claim 7, <u>Matena</u> further teaches wherein the control logic is provided to generate an instruction to start, terminate or restart a particular process executed server nodes based on a command received from a remote device (para. [0139], [0208]-[0212], and [0231]-[0248]).

As to Claim 8, <u>Matena</u> does not explicitly teach wherein the communication interface further comprises a named pipe to send and receive commands between the control logic and the launch logic. However, it is well known in the art there are multiple methods to establish communication between processes in the network such as pipes, FIFOs, Stream and Messages, Message Queues, Shared Memory, etc. It would have been obvious to one of ordinary skill in the art at the time of invention to modify and improve the system of <u>Matena</u> and Sohda to include named pipe to send and receive commands between the control logic and the launch logic, thus, applying well known technique in the art to the system instead developing a new one.

As to Claim 9, <u>Matena</u> further teaches wherein the control logic comprises: a signal handler to receive and interpret signals from a management console (para. [0220]-[0223]) (Subscribing to events inherently requires registering a message handler to receive the sent event messages).

As to Claim 10, Matena further teaches wherein the control logic comprises: a server connector to enable connection with an external server (para. [0417] and [0421]).

As to Claim 13, Matena discloses the invention substantially as claimed including a method comprising:

a) executing Java processes for a plurality of server nodes in an instance (108, Fig. 1);

b) obtaining status regarding the Java processes executed by the server nodes in the instance (para. [0214]);

c) storing the status regarding the Java processes in a communication interface (para. [0216]), the communication interface updating and maintaining the status in a memory (JAC 2401 maintains … belong to this container group; page 16, paragraphs [0216]-[0218]);

d) accessing the status in the communication interface (para. [0214] and [0220]-[0223]).

Matena does not explicitly teach shared memory. However, Sohda teaches shared memory between nodes in clusters (Abtract and pages 165-167, sections 2.2 "Summary of JSDM Implementation Choices/Policies", section 3 "JDSM Implementation and section 3.2 "DSM Runtime Implementation").

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Matena with the teachings of Sohda by using shared memory instead of memory thus enable multiple server nodes to access the shared memory as needed.

As to Claim 14, Matena further teaches enabling control of the Java processes based on an instruction received from a remote device (para. [0220]-[0223]) (The "JAC API" allows for control of an application, thereby meeting this claim limitation).

As per Claims 11, 12 and 15, these claims are rejected for the same reasoning as applied to Claim 4 above.

As to Claim 16, Matena further teaches;

a) detecting a failure of a process within the cluster by accessing the status in the communication interface (para. [0242]-[0248]); and

b) restarting the failed process (para. [0242]-[0248]).

As to Claim 17, Matena discloses the invention substantially as claimed including a machine-readable medium that provides instructions, which when executed by a processor cause the processor to perform operations comprising:

a) executing Java processes for a plurality of server nodes in an instance (108, Fig. 1);

b) obtaining status regarding each of the Java processes executed by the server nodes in the instance (para. [0214] and [0220]-[0223]); and

c) storing the status regarding the Java processes into a memory (para. [0214], [0220]-[0223], and [0413]).

Matena does not explicitly teach shared memory. However, Sohda teaches implementing shared memory between nodes in clusters (Abtract and pages 165-167, sections 2.2 "Summary of

JSDM Implementation Choices/Policies", section 3 "JDSM Implementation and section 3.2 "DSM Runtime Implementation").

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Matena with the teachings of Sohda by using shared memory instead of memory thus enable multiple server nodes to access the shared memory as needed.

As per Claim 18, being directed to a machine-readable medium encoded with instructions to perform the steps of the system of Claim 4, this claim is rejected for the same reasoning as applied to Claim 4.

As to Claim 19, Matena further teaches wherein the operations performed by the processor further comprise:

a) receiving instructions via a communication interface (para. [0139], [0208]-[0212], and [0231]-[0248]); and

b) starting, terminating or restarting a process based on the instructions received via the communication interface (para. [0139], [0208]-[0212], and [0231]-[0248]).

As to Claim 20, Matena as modified by Sohda further teaches wherein the operations further comprise: detecting a failure of a process within the cluster by accessing the status in the shared memory and automatically restarting the failed process (see Matena: para. [0139], [0208]-[0212], [0231]-[0248], and [0413]) and (discussion in claim 17 above regarding Sohda teaching shared memory).

As to Claim 21, Matena discloses the invention substantially as claimed including an apparatus comprising:

a) a cluster having a first instance and a second instance, each of the first and second instances including a plurality of server nodes (para. [0078], [0083]);

b) a control logic to start each respective instance by initiating a launch logic for each respective server node in the first and second instances (para. [0114], [0123]-[0132], and [0413]);

c) the launch logic, for each respective server node in the first and second instances, to further launch Java processes, and obtain a status of the Java processes to store and maintain in a memory (para. [0209], [0216]-[0218]);

d) and the control logic to access the status obtained by the launch logic (para. [0221], [0242]-[0248]).

Matena does not explicitly teach shared memory. The only different between this claim and the system of Matena is that in Matena, the control logic obtains the status of the processes using message passing, wherein in this claim, the control logic obtains the status of the processes using shared memory. However, Sohda teaches shared memory between nodes in clusters that implemented in Java (Abtract and pages 165-167, sections 2.2 "Summary of JSDM Implementation Choices/Policies", section 3 "JDSM Implementation and section 3.2 "DSM Runtime Implementation").

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Matena with the teachings of Sohda by using shared memory technique instead of message passing because of all well-known advantages of using shared memory over message passing in the art, and both the control logic and launch logic can access and update the status of the processes concurrently. Furthermore, using shared memory or message passing are design choice.

As to Claim 22, Matena as modified by Sohda teaches a shared memory to enable exchange of information between the Java processes and the control logic (see discussion in claim 21 above regarding Sohda teaches implementing shared memory in the cluster to allow multiple nodes share information).

As to Claim 23, Matena further teaches wherein the launch logic loads a virtual machine and executes Java processes (para. [0193]-[0196]).

As per Claim 24, being directed to an apparatus performing substantially the same function as the system of Claim 4, this claim is rejected for the same reasoning as applied to Claim 4.

As to Claim 25, Matena further teaches wherein the control logic detects a failure of a process within the cluster; and automatically restarts operations of the failed process (para. [0139], [0208]-[0212], and [0231]-[0248]).

As to Claim 26, Matena further teaches a signal handler to receive a command from a remote device and controlling one of the Java processes based on the command received from the remote device (para. [0220]-[0223]) (Subscribing to events inherently requires registering a message handler to receive the sent event messages).

As per Claim 27, being directed to an apparatus performing substantially the same function as the system of Claim 8, this claim is rejected for the same reasoning as applied to Claim 8.

As to Claim 28, Matena discloses the invention substantially as claimed including a system comprising:

a) a cluster having a first instance and a second instance, each of the first and second instances including a plurality of server nodes (para. [0078], [0083]);

b) means for starting each instance by executing Java processes in each respective server node (para. [0114], [0123]-[0132], [0209] and [0413]); and

c) means for enabling exchange of information (para. [0221] that is stored and maintained in a memory between the Java processes and the means for starting each instance (para. [0208]-0209, [0216]-[0218] and [0221]).

Matena does not explicitly teach shared memory. However, Sohda teaches shared memory between nodes in clusters that implemented in Java (Abtract and pages 165-167, sections 2.2 "Summary of JSDM Implementation Choices/Policies", section 3 “JDSM Implementation and section 3.2 “DSM Runtime Implementation”).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Matena with the teachings of Sohda by using shared memory instead of just memory thus other nodes in the cluster can also access the shared memory as needed.

As to Claim 29, Matena further teaches a means for loading a virtual machine and execute a Java process in the virtual machine (para. [0193]-[0196]).

As to Claim 30, Matena as modified by Sohda further teaches wherein the means for enabling exchange of information comprises: a shared memory having a plurality of entries (see discussion in claim 28 above regarding teaching of Sohda).

As to Claim 31, Matena as modified by Sohda further teaches means for obtaining status for each of the Java processes; and means for updating the shared memory with the obtained status (see Matena: para. [0214] and [0220]-[0223]) and (discussion in claim 28 above regarding teaching of shared memory by Sohda).

As to Claim 32, Matena as modified by Sohda further teaches:

a) means for accessing the shared memory to monitor the status of each of the Java processes (see Matena: para. [0413]) and (discussion in claim 28 above regarding teaching of shared memory by Sohda); and

b) means for sending an instruction to the launch means to start, terminate or restart a particular process executed in the cluster (para. [0139], [0208]-[0212], and [0231]-[0248]).

As to Claim 33, Matena as modified by Sohda further teaches:

a) means for enabling a user to monitor and control the Java processes running in the cluster from a management console coupled to the means for controlling (para. [0214] and [0220]-[0223]); and

b) means for enabling a connection with an external server (para. [0417] and [0421]).

As to Claim 34, Matena as modified by Sohda further teaches wherein the storage of information in the shared memory is done independent of the accessing of that information (see Sohda: page 167, section "3.2 DSM Runtime Implementation").

As to Claim 35, Matena as modified by Sohda further teaches wherein a persistent data structure is stored in the shared memory to enable an independent exchange of information (see Sohda: page 167, section "3.2 DSM Runtime Implementation").

As to Claim 36, this claim is rejected for the same reasoning as applied to Claims 13 and 34, above.

As to Claim 37, this claim is rejected for the same reasoning as applied to Claims 17 and 34, above.

As to Claim 38, this claim is rejected for the same reasoning as applied to Claims 21 and 35.

As to Claim 39, this claim is rejected for the same reasoning as applied to Claim 34 above.

Response to Arguments

7. Applicant's arguments with respect to claims 1-39 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO 892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DIEM K. CAO whose telephone number is (571)272-3760. The examiner can normally be reached on Monday - Friday, 7:30AM - 3:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DC
July 1, 2008

/Li B. Zhen/
Primary Examiner, Art Unit 2194